Universidade Federal do Rio de Janeiro

Instituto Tércio Pacitti de Aplicações e

Pesquisas Computacionais

Ítalo Fernandes Aguiar

PROPOSTA DE UTILIZAÇÃO DA FERRAMENTA *ZABBIX* NO GERENCIAMENTO DE REDES:

Um Estudo de Caso no Ambiente da FAB Segundo Boas Práticas de Governança de TI

Rio de Janeiro

2013

Ítalo Fernandes Aguiar

PROPOSTA DE UTILIZAÇÃO DA FERRAMENTA *ZABBIX* NO GERENCIAMENTO DE REDES:

Um Estudo de Caso no Ambiente da FAB Segundo Boas Práticas de Governança de TI

Monografia apresentada para obtenção do título de Especialista em Gerência de Redes de Computadores no Curso de Pós-Graduação Lato Sensu em Gerência de Redes de Computadores e Tecnologia Internet do Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais da Universidade Federal do Rio de Janeiro – NCE/UFRJ.

Orientador:

Moacyr Henrique Cruz de Azevedo, M.Sc., UFRJ, Brasil

Rio de Janeiro

2013

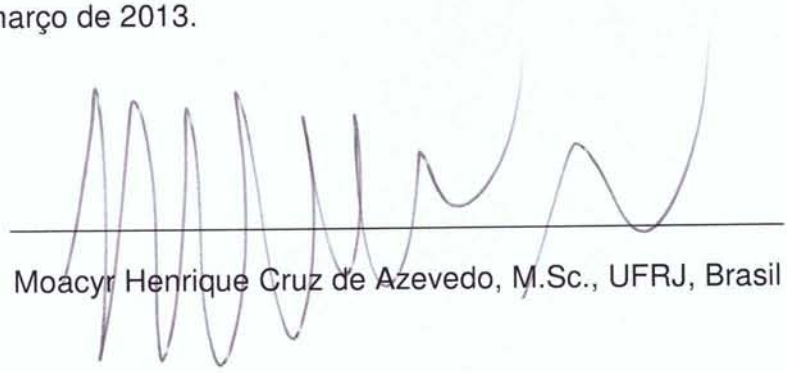
Ítalo Fernandes Aguiar

PROPOSTA DE UTILIZAÇÃO DA FERRAMENTA *ZABBIX* NO GERENCIAMENTO DE REDES:

Um Estudo de Caso no Ambiente da FAB Segundo Boas Práticas de Governança de TI

Monografia apresentada para obtenção do título de Especialista em Gerência de Redes de Computadores no Curso de Pós-Graduação Lato Sensu em Gerência de Redes de Computadores e Tecnologia Internet do Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais da Universidade Federal do Rio de Janeiro – NCE/UFRJ.

Aprovada em março de 2013.



Moacyr Henrique Cruz de Azevedo, M.Sc., UFRJ, Brasil

AGRADECIMENTOS

Agradeço a todos que me apoiaram nessa longa jornada que começou desde a graduação na faculdade até este presenta momento, em que completo minha monografia de Pós-Graduação em nível de Especialização.

Grande importância teve, nessa jornada, o pessoal do Centro de Computação de Aeronáutica do Rio de Janeiro, que me incentivou a ingressar neste curso, portanto, muito dessa conquista deve-se a eles.

Finalmente, agradeço a minha namorada e companheira, que esteve ao meu lado durante grande parte dessa trilha, sempre me apoiando nos momentos bons e ruins.

RESUMO

AGUIAR, Ítalo Fernandes. **PROPOSTA DE UTILIZAÇÃO DA FERRAMENTA *ZABBIX* NO GERENCIAMENTO DE REDES: Um Estudo de Caso no Ambiente da FAB Segundo Boas Práticas de Governança de TI.** Monografia (Especialização em Gerência de Redes e Tecnologia Internet). Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2013.

Estudo de caso sobre implantação de um sistema eficiente de monitoramento e gerenciamento de serviços de rede em um ambiente corporativo da Força Aérea Brasileira.

Foi definida a utilização do Zabbix dentre as opções de sistema para o gerenciamento. A sua implantação e operação foi baseada em alguns princípios do CobiT (*Control Objectives for Information and related Technology*), pois este modelo tem como foco as boas práticas amplamente reconhecidas na área de Governança de TI, a fim de garantir o alinhamento da equipe de TI com área de negócios.

ABSTRACT

AGUIAR, Ítalo Fernandes. **PROPOSTA DE UTILIZAÇÃO DA FERRAMENTA *ZABBIX* NO GERENCIAMENTO DE REDES: Um Estudo de Caso no Ambiente da FAB Segundo Boas Práticas de Governança de TI.** Monografia (Especialização em Gerência de Redes e Tecnologia Internet). Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2013.

Case study on the implementation of an efficient system of monitoring and management of network services in a corporate environment of the Brazilian Air Force.

It was defined using the Zabbix system among the options for management systems. Its implementation and operation was based on some principles of CobiT (Control Objectives for Information and related Technology), because this model focuses on best practices widely recognized in the area of IT governance in order to ensure alignment of IT staff with businesses executives.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AI	Acquire and Implement (Adquirir e Implementar)
BD	Banco de Dados
CCA	Centro de Computação de Aeronáutica
CCA-RJ	Centro de Computação de Aeronáutica do Rio de Janeiro
CobiT	Control Objectives for Information and related Technology (Objetivos de Controle para Tecnologia da Informação e Afins)
DTI	Diretoria de Tecnologia da Informação
DNS	Domain Name System (Sistema de Nome de Domínio)
DS	Deliver and Support (Entregar e Suportar)
FAB	Força Aérea Brasileira
FTP	File Transfer Protocol (Protocolo de Transferência de Arquivo)
ME	Monitor and Evaluate (Monitorar e Avaliar)
OC	Objetivos de Controle (*Control Objectives – CO*)
PO	Plan and Organize (Planejar e Organizar)
RACI	Responsible, Accountable, Consulted and Informed (Responsável, Responsabilizado, Consultado e Informado)
SAUTI	Serviço de Atendimento aos Usuários de TI
SIGPES	Sistema de Informações Gerenciais de Pessoal
SILOMS	Sistema Integrado de Logística de Material e de Serviços
SMTP	Simple Mail Transfer Protocol (Protocolo de transferência de correio simples)
TI	Tecnologia da Informação

SUMÁRIO

1 **INTRODUÇÃO**

No decorrer deste capítulo, serão tratadas questões sobre tematização, motivação, contextualização e especificação de requisitos para o desenvolvimento posterior dos assuntos a serem abordados neste Trabalho de Pós-Graduação Lato Sensu.

1.1 TEMATIZAÇÃO

As redes de computadores, que no começo não passavam de uma pilha de protocolos desenvolvidos para troca de textos acadêmicos entre universidades, hoje já faz parte da vida de várias faixas etárias de uma expressiva parte da sociedade em todo o mundo. Além de grandes avanços nas áreas científicas e acadêmicas, ela propiciou inovações para o comércio, jornalismo e relações interpessoais, por exemplo.

Porém, lado a lado com essa face positiva da utilização das redes de computadores, esta traz consigo de forma intrínseca uma série de preocupações que devem ser levadas em consideração. Justamente por serem formadas por uma séria de protocolos e suportarem uma grande diversidade de serviços nos ambientes empresariais, as redes de computadores têm de lidar constantemente com situações em que falhas podem acontecer, colocando em cheque toda a estrutura de serviços suportados por estas.

Essa preocupação tem de ser traduzida em uma política de gerenciamento de rede capaz de monitorar desde simples alterações em um sistema operacional de um computador isolado na rede interna até oscilações na qualidade de largura de banda em alguns dos *links* de internet que venham a suportar um ambiente empresarial. Tudo isso preferencialmente de forma clara em tempo hábil, do contrário torna-se

impraticável uma reação efetiva contra possíveis falhas nesse ambiente corporativo.

1.2 CONTEXTUALIZAÇÃO E MOTIVAÇÃO

A Força Aérea Brasileira – FAB, como qualquer outra grande empresa, enxergou na utilização de computadores interligados em redes uma tecnologia que poderia trazer muitas vantagens. Assim, o Comando da Aeronáutica - COMAER investiu na elaboração da INTRAER, a rede de computadores interna (ou intranet) da FAB.

Na INTRAER, as redes de dados estão distribuídas por regiões, representadas pelas diversas Organizações Militares – OMs da FAB. Atualmente, essa rede está distribuída em regiões nacionais além de alguns pontos internacionais de projetos e programas de intercâmbio de informações.

Cada OM tem sua própria estrutura de rede interna e é responsável por realizar seu gerenciamento, de forma que seus servidores e serviços possam ser acessados por meio de toda a INTRAER.

Além da facilidade de comunicação entre as máquinas localizadas dentro da INTRAER, via de regra, em cada OM existe um portal de comunicação (*gateway)* que permite o acesso de usuários internos a Internet.

Somado a essa estrutura, existem ainda três Centros de Computação de Aeronáutica – CCA, subordinados à Diretoria de Tecnologia da Informação – DTI [DTI, RICA 21-236, 2011, p. 23], órgão que tem por finalidade normatizar, planejar, implantar, coordenar, controlar e fiscalizar as atividades relativas à tecnologia da informação do Comando da Aeronáutica [DTI, RICA 21-236, 2011, p. 9].

Estes Centros atuam nas diversas áreas relacionadas a TI, desenvolvendo sistemas de informação e projetos de rede, executando o suporte aos sistemas

corporativos e demais atividades de TI da FAB. Para este trabalho, será tomado como estudo de caso mais especificamente o CCA-RJ (Centro de Computação de Aeronáutica do Rio de Janeiro), localizado na Ilha do Governador no estado do Rio de Janeiro.

A INTRAER, com toda sua complexa estrutura espalhada por todo o país, tem como, dentre vários propósitos, o de interconectar as suas OMs mais distantes aos serviços disponíveis no CCA-RJ, de forma que estas Organizações possam “consumir” serviços essenciais como, por exemplo, SIGPES (Sistema de Informações Gerenciais de Pessoal) e SILOMS (Sistema Integrado de Logística de Material e de Serviços).

Para prestar apoio aos usuários de TI da FAB, esta implantou, também no CCA-RJ, o SAUTI (Serviço de Atendimento aos Usuários de Tecnologia da Informação), atuando como uma central de serviços e prestando atendimento por meio de chamados, sejam estes on-line ou via contato telefônico.

A preocupação da FAB com o monitoramento efetivo de suas redes em toda a extensão do território nacional pode ser observada não só nas operações que tomam corpo no dia-a-dia dos CCAs, mas também na atenção que é dispensada a esse assunto nos cursos que estes centros prestam a todas as OMs do país e cujo conteúdo programático foca na utilização de *softwares* de monitoramento e gerência [DTI, ICA 37-449, 2011, p.10].

Dessa forma, dadas as proporções e heterogeneidade no ambiente corporativo de TI na FAB, ao desenvolver qualquer projeto de implantação de novo software, faz-se necessário atentar para os princípios básicos de Governança de TI, que consistem basicamente em um conjunto de práticas, padrões e relacionamentos estruturados, assumidos por gestores, técnicos e usuários de TI de uma organização, com a

finalidade de garantir controles efetivos, minimizar os riscos, ampliar o desempenho, otimizar a aplicação de recursos, suportar as melhores decisões e consequentemente alinhar TI aos negócios.

Portanto, para se atender aos requisitos de gerenciamento de rede exigidos no contexto referenciado neste Trabalho de Pós-Graduação, torna-se necessário a implantação, seguindo boas práticas de Governança de TI, de uma política de gerenciamento de disponibilidade e performance aplicada a um conjunto de ativos e serviços de rede.

1.3 ESPECIFICAÇÃO DE REQUISITOS

Nesta seção, serão discutidos os fatores envolvidos no contexto do problema estudado neste trabalho, como efeitos adversos observados, possíveis causas, tarefa proposta e qual o propósito desta. A partir daí, torna-se possível enunciar o problema com clareza e posteriormente definir, dentre soluções analisadas, a ideal para este.

1.3.1 **Efeitos Adversos Observados**

Problemas de disponibilidade e qualidade de serviços ou de *links* têm sido constantemente relatados ao SAUTI quando o problema já atinge uma quantidade relevante de usuários, trazendo uma série de complicações a atividades fins da Força Aérea, como controle de missões de vôo e gestão de suprimentos para militares na Amazônia.

Além disso, em alguns sites disponibilizados pela FAB à sociedade brasileira, observa-se a vulnerabilidade real com relação a técnicas de invasão como a desfiguração de *website* [ESTADAO, 2008]. Estas consistem em ataques nos quais a

aparência do site é modificada por *crackers*, indivíduos que praticam a quebra de um sistema de segurança, cujas alterações não são detectadas em tempo hábil para que seja tomada alguma contramedida.

1.3.2 **Causas**

Dentre as principais causas dos efeitos adversos apresentados, podem ser citadas as seguintes:

- Falhas não detectadas (em tempo hábil) de disponibilidade, performance e integridade tanto dos sistemas corporativos como de serviços de suporte como DNS, SMTP, Servidor WEB e FTP;
- Ausência de controle eficiente de configuração de ativos e servidores;
- Falhas na comunicação eficiente de mudanças de regras de roteamento e *firewalls* entre as partes responsáveis pelos sistemas corporativos e os usuários finais, ocasionando demora na localização e isolamento do problema, para posterior solucionamento;
- Inexistência de histórico de disponibilidade e performance de serviços, o que impossibilita comparação no surgimento de situações instáveis com as operacionais.

Mesmo quando os problemas acima são monitorados por *softwares* específicos para tais finalidades, estes não são implantados de uma forma sistemática que siga as boas práticas de governança de TI, de forma que se torna impraticável ao administrador de rede conseguir analisar, diagnosticar, priorizar e reportar todos os problemas de forma eficiente e tomar assim alguma medida reativa em tempo hábil.

1.3.3 **Tarefa**

Estudar a utilização de ferramentas que implementem uma solução técnica de gerenciamento de redes operando no contexto da INTRAER. Tal solução, além de atacar as causas dos problemas citados no item anterior, deve ser implantada de maneira sistemática, seguindo um conjunto de procedimentos considerados amplamente como boas práticas no contexto da governança de TI

1.3.4 **Definição de Alternativas de Soluções**

A **Alternativa 1** para o problema definido seria terceirizar a resolução deste para empresas especializadas no ramo de gerenciamento de ativos e serviços de redes.

Uma **Alternativa 2** seria adquirir separadamente softwares que realizam as funções de monitoramento de *link*; controle de inventário de ativos de rede e de servidores; monitoramento de disponibilidade, performance e integridade de serviços e servidores com possibilidade de geração de diversos tipos de relatórios e notificações; e armazenamento de informações (Banco de Dados – BD) para geração de histórico. Adquiridos os softwares, teria de se desenvolver uma solução de integração de todos eles para finalmente poder colocar em prática todo o processo de gerenciamento de rede requerido.

Como **Alternativa 3** tem-se a utilização de uma categoria de *software* de código aberto *(open source)*, que atua como uma solução (técnica) de apoio na consolidação de um sistema que aborde de forma integrada todos os quesitos da Alternativa 2.

A Alternativa 3 aborda mais especificamente a utilização do *software* Zabbix, cuja escolha será mais bem analisada no **Capítulo 3**.

1.3.5 **Análise de Viabilidade das Alternativas de Solução**

A Análise realizada baseia-se em verificar, para cada Alternativa de Solução, os seguintes requisitos:

- **Adequabilidade**, que analisa se são criadas as condições necessárias para a concretização da tarefa, permitindo que seja plenamente atingido o propósito proposto;
- **Praticabilidade** do esforço requerido, que analisa os meios disponíveis de implementação da solução em confronto com as dificuldades para tal.
- **Aceitabilidade**, que verifica se os benefícios auferidos com a implantação da solução compensam o esforço a ser despendido e o risco a se correr.

Dessa forma, temos para as Alternativas apresentadas:

a) **Alternativa 1**: considerada inadequada, pois, se por um lado existem chances reais de as causas do problema proposto serem devidamente combatidas, por outro se depara com dois novos problemas:

- Transferência inexistente ou apenas parcial de conhecimento na realização da tarefa proposta; e
- Risco devido à necessidade de ceder a terceiros informações sigilosas no que diz respeito à INTRAER.

Além disso, é considerada parcialmente praticável, pois pode envolver altos investimentos financeiros para a manutenção do serviço terceirizado com o passar do tempo.

b) **Alternativa 2**: considerada impraticável devido aos seguintes fatores:

- Alto investimento em recursos humanos para capacitação na operação de todas as ferramentas adquiridas separadamente e no desenvolvimento da integração eficiente de todas estas; e
- Caso houvesse alguma ferramenta de fornecimento não gratuito, haveria necessidade de investimentos financeiros adicionais.

c) **Alternativa 3**: considerada adequada, pois a equipe de TI da FAB deteria o conhecimento necessário para realizar a tarefa proposta e não haveria necessidade de ceder informações sigilosas da INTRAER a terceiros. Essa Alternativa também é considerada praticável, pelo fato de o Zabbix já constituir uma solução pronta. O foco da capacitação de pessoal seria no gerenciamento apenas dessa plataforma. Além disso, como já citado anteriormente, o Zabbix é código aberto e não exige investimento financeiro em sua aquisição.

Finalizando, além de adequada e praticável, a Alternativa 3 é considerada aceitável, pois, no contexto proposto neste trabalho, o benefício obtido por meio de sua implementação (adequabilidade) compensa o esforço requerido, que envolve tão somente investimento na capacitação de pessoal (praticabilidade).

1.3.6 **Definição da Alternativa de Solução Escolhida**

Como a Alternativa 3 foi a única a atender a todos os critérios propostos na Análise de Viabilidade, torna-se a Alternativa de Solução Escolhida para este Trabalho de Pós-Graduação e, a partir dela, este trabalho de pesquisa será desenvolvido.

1.4 INTITULAÇÃO

Desta forma, de acordo com a alternativa de solução escolhida, este Trabalho de Graduação pode ser intitulado como:

“**Proposta De utilização da Ferramenta *Zabbix* no Gerenciamento de Redes: Um Estudo de Caso no Ambiente da FAB Segundo Boas Práticas de Governança de TI”**

2 **ZABBIX COMO SOLUÇÃO TÉCNICA NO GERENCIAMENTO DE ATIVOS DE REDES E SERVIDORES**

No decorrer deste capítulo, serão tratadas questões sobre Conceitos, Requerimentos, Instalação e Configuração da ferramenta Zabbix.

2.1 CONCEITOS

Zabbix é uma ferramenta open source de monitoramento de disponibilidade e performance que oferece monitoração avançada, sistemas de visualização e alertas dentre os quais podem ser citados [ZABBIX SIA, 2013]:

- Descoberta automática de servidores e ativos de rede;
- Descoberta baixo nível (relativa a descoberta de interfaces de rede, dispositivos de armazenamento dentre outros);
- Monitoração distribuída com administração centralizada via *web*;
- Suporte para monitoramento ativo e passivo, conhecidos respectivamente como *pooling* e *trap*;
- Agentes nativos de monitoramento para diversas plataformas, como Linux, Solaris, HP-UX, FreeBSD e as da família Windows;
- Trabalha com ferramentas livres como Linux, Apache e PHP;
- Armazena, de forma nativa, todas as informações (configurações e dados de disponibilidade e performance) em bancos de dados relacionais livres, como PostgreSQL e MySQL;
- Notificação flexível (painel de alertas, mensagens de *e-mail* e até mesmo SMS); e

- *Logs* de auditoria.

2.2 FERRAMENTAS ALTERNATIVAS

Dentro do contexto da alternativa 3, a solução que mais se aproxima de atingir todos os requisitos exigidos nesta alternativa é o Nagios [NAGIOS, 2013].

Este traz muitas das vantagens encontradas no Zabbix, porém, após uma baterias de testes de utilização do Nagios em um dos segmentos de redes no CCA-RJ, constatou-se que ele não se apresenta como uma solução totalmente viável devido aos seguintes fatos:

- Integração com banco de dados relacionais (MySQL e PostgreSQL) é um processo não muito bem estabelecido;
- Sua capacidade de monitorar *link* está amarrada a um período de amostragem com precisão limitada em 5 minutos. Quaisquer alterações ocorridas em um período menor do que esse passam despercebidas para o Nagios;
- Sua forma de gerenciamento é, a priori, via console. *Plugins* para habilitar configuração via *web*, como Nconfig e NagiosQL, se demonstraram instáveis em testes prévios feito em ambiente controlado no CCA-RJ;
- O pacote núcleo do Nagios tem apenas as funções mais essenciais, de forma que este, para atingir uma variedade de formas de monitoramento, tem de ser integrado a diversos *plugins*, sendo muitos destes feitos por comunidades independentes e sem garantia. E, mais grave ainda do que isso, muitos desses *plugins* vêm sendo descontinuados, o que traz um risco intrínseco e indesejável.

Dessa forma, somando os conceitos apresentados do Zabbix e levando em conta as limitações apresentadas para o Nagios, definiu-se, para o estudo de caso

desenvolvido para este Trabalho, que será utilizada a ferramenta Zabbix.

2.3 ARQUITETURA

A arquitetura Zabbix pode ser caracterizada como um sistema de monitoramento semi-distribuído com gerenciamento centralizado. Enquanto algumas instalações têm um banco de dados central, é possível usar monitoramento distribuídos com nós e *proxies* [OLUPS, 2010, p.9].

A figura 1 trata de uma configuração relativamente comum mas que possibilita uma gama extensa de capacidades de monitoramento em uma rede composta por equipamentos heterogêneos.

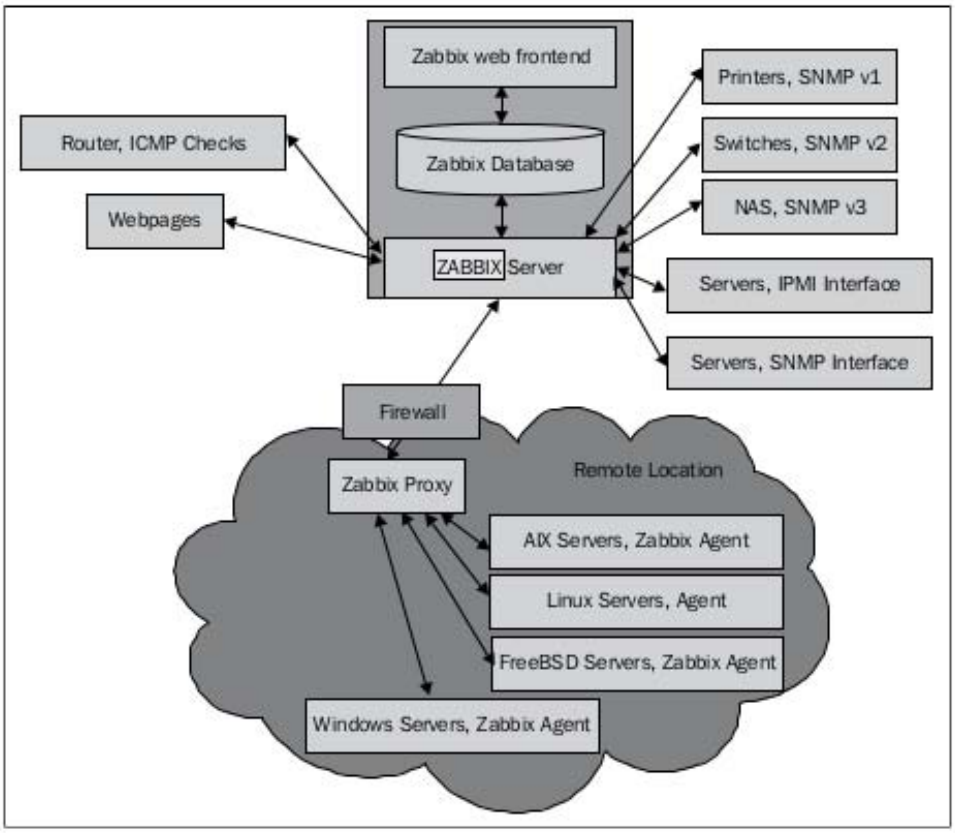


Figura 1 - Arquitetura convencional em um Sistema de monitoramento ZABBIX

Alguns dos principais componentes do Zabbix são descritos a seguir [ZABBIX SIA, 2013b]:

- **Servidor Zabbix (*Zabbix Server*)**: componente central para quem os agentes reportam informação de disponibilidade e integridade, bem como estatísticas.
- **Banco de Dados (*Zabbix Database*)**: um repositório central, onde todas as configurações, informações estatísticas e operacionais são armazenadas;
- **Interface Web (*Zabbix Web Frontend*)**: para simplificar o acesso ao Zabbix de qualquer lugar e de qualquer plataforma, é disponibilizada de forma nativa uma interface *web*, que normalmente (mas não necessariamente) roda na mesma máquina física que o Servidor Zabbix;
- **Zabbix Proxy**: coleta informação de disponibilidade e performance como se fosse o servidor Zabbix, passando para este, em momento oportuno, as informações coletadas. O *proxy* é uma parte opcional do Zabbix que atua trazendo benefício direto para distribuição de carga em relação a uma arquitetura com apenas um servidor central;
- **Agente Zabbix (Zabbix Agent)**: são implantados diretamente nas máquinas alvo, para monitorar recursos locais (memória RAM, discos, usuários, interface de rede, dentre outros) e aplicações, reportando a informação coletada ao Servidor Zabbix;
- **Fluxo de informações**: Um conjunto de itens principais são utilizados no fluxo geral de informações que se desenrola em um sistema gerenciado com a ferramenta Zabbix [ZABBIX SIA, 2013c]. Dentre esses itens, merecem destaque:
    - ***host***: um dispositivo na rede a ser monitorado;
    - ***host_group***: um grupo lógico de *hosts*, que pode conter, além destes,

um conjunto de *templates*;

- **item**: parte elementar de informação que é extraída de um *host* seguindo uma métrica definida. Como exemplo, podem ser citados: tráfego de entrada em uma interface de rede e *check* de resposta a *ping*;
- ***trigger***: uma expressão lógica que define uma situação limite, usada para avaliar as informações recebidas relativas a cada item. Se uma informação recebida extrapola o limite definido, o *trigger* passa do seu estado “Ok” para o estado “*Problem*”, caso contrário, o *trigger* continua em “Ok”;
- ***event***: a simples ocorrência de algo que mereça atenção, como, por exemplo, uma mudança de estado de um *trigger*;
- ***action***: um procedimento pré-definido para reagir a um *event*;
- ***media***: um meio de enviar *notifications*;
- ***notification***: uma mensagem sobre algum evento enviada para algum usuário Zabbix via canal media pré-definido;
- ***template***: um conjunto de entidades (*items, triggers, graphs, screens* dentre outros) prontas para ser aplicado a um ou mais *hosts*. Sua função é acelerar a configuração de tarefas de monitoramento em um *host*, facilitando a aplicação de mudanças em massa;
- ***application***: um agrupamento lógico de itens;
- ***web scenario***: uma ou mais requisições HTTP para checar a disponibilidade de um *web site*.

2.4 REQUERIMENTOS

Os principais requisitos para instalação do Zabbix são divididos em *Hardware*, Plataforma e Banco de Dados, conforme explicitado a seguir [ZABBIX SIA, 2013d].

2.4.1 ***Hardware***

Quanto à memória RAM, um ponto de partida seria ter no mínimo 128MB, mas tudo vai depender do tamanho do ambiente a ser monitorado, o que impactará diretamente no tamanho do banco de dados, que exigirá mais ou menos memória RAM para ter uma performance aceitável. O mesmo raciocínio funciona para recursos de CPU.

Como exemplo de configurações e suas respectivas capacidades de monitoramento, tem-se a Tabela 1.

Tabela 1 - Relação entre Configurações e Capacidades

Nome	**Plataforma**	**CPU/Memória**	**Banco de Dados**	**Número de *Hosts***
Pequeno	Ubuntu Linux	PII 350MHz 256MB	SQLite	20
Medio	Ubuntu Linux 64 bit	AMD Athlon 3200+ 2GB	MySQL InnoDB	500
Extenso	Ubuntu Linux 64 bit	Intel Dual Core 6400 4GB	RAID10 MySQL InnoDB or PostgreSQL	>1000
Muito Extenso	RedHat Enterprise	Intel Xeon 2xCPU 8GB	Fast RAID10 MySQL InnoDB or PostgreSQL	>10000

2.4.2 **Plataforma**

Devido a requisitos de segurança e a natureza crítica do servidor de monitoramento, UNIX é o único sistema operacional que consegue, de forma

consistente, garantir a performance, tolerância a falhas e resiliência necessárias.

Vale ressaltar que o agente Zabbix foi devidamente validado para ser executado nas seguintes plataformas:

- Linux;
- IBM AIX;
- FreeBSD;
- NetBSD;
- OpenBSD ;
- HP-UX ;
- Mac OS X ;
- Solaris; e
- Windows: 2000, Server 2003, XP, Vista, Server 2008, W7 (Zabbix agent apenas).

2.4.3 **Banco de Dados**

Os tipos de banco de dados habilitados para trabalhar com o Zabbix são:

- MySQL 5.0 ou mais recente;
- Oracle 10g ou mais;
- PostgreSQL 8.1 ou mais; e
- SQLite 3.3.5 ou mais.

O tamanho do banco de dados para trabalhar com o Zabbix vai depender de uma série de variáveis. Tal dependência pode ser resumida nas equações da Tabela 2.

Tabela 2 - Tabela base para cálculo do tamanho do banco de dados do servidor

Parâmetro	Espaço Requerido (em bytes)
Configuração	Tamanho fixo (normalmente 10MB)
Histórico	**dias*(itens/taxa de atualização)*24*3600*bytes** itens: número de itens dias: número de dias para manter histórico taxa de atualização: taxa média de atualização para itens bytes: normalmente 50 bytes.
Trends	**dias*(itens/3600)*24*3600*bytes** itens: número de itens dias: número de dias para manter histórico bytes: normalmente 128 bytes.
Eventos	**dias*eventos*24*3600*bytes** eventos: número de eventos por segundo (1, no cenário de pior caso) dias: número de dias para manter histórico bytes: normalmente 130 bytes.

Assim, tem-se que, para o disco, o tamanho total requerido para comportar os dados que serão capturados ao longo do tempo pode ser calculado como:

Tam_total = Configuração + Histórico + Trends + Eventos

2.5 INSTALAÇÃO

O processo de instalação do Zabbix consiste principalmente de duas etapas: Instalação do servidor e Instalação dos agentes nas máquinas a serem monitoradas, de acordo com o explicitado a seguir.

2.5.1 **Instalação do Servidor**

Para a instalação do servidor, após uma série de tentativas e erros, chegou-se em um denominador comum que pode ser resumido na sequencia de passos apresentada no Anexo 1.

2.5.2 **Instalação do Agente Zabbix**

Para a instalação do agente Zabbix, também foi feita uma pesquisa das melhores formas de se instalar automaticamente o agente. Foi desenvolvido um *script* focado para a distribuição Debian 6.0, mas que pode facilmente ser adaptado para demais distribuições UNIX.

A instalação para Windows é trivial e para cada versão do Windows pode ser encontrada facilmente tutoriais na internet, por isso essa etapa não foi considerada neste Trabalho. O *script* desenvolvido para o Debian encontra-se representado no Anexo 2.

3 **BOAS PRÁTICAS EM GOVERNANÇA DE TI**

A Governança de TI constitui basicamente um conjunto de melhores práticas assumidas por todos os atores da instituição, com intuito de garantir controles efetivos, minimizar riscos, ampliar o desempenho, otimizar a aplicação de recursos e orientar as decisões [SERPRO, 2013].

A premissa mais importante da Governança de TI é alinhar as diretrizes e objetivos estratégicos no campo da tecnologia da informação, com o desafio de aumentar a eficiência e a produtividade. Em linhas gerais, o documento se propõe a identificar com clareza as seguintes questões:

- Que decisões devem ser tomadas para garantir a gestão e o uso eficaz da TI?
- Quem deve tomar essas decisões?
- Como tomá-las e monitorá-las?

A Governança de TI compreende vários mecanismos e componentes que, logicamente integrados, permitem o desdobramento da estratégia de TI até a operação dos produtos e serviços correlatos. Dentre as melhores práticas, podem ser citadas: PMBOK, CobiT, ITIL, e CMMI.

Dessa forma, para o propósito deste trabalho, serão analisadas e utilizadas algumas das abordagens especificadas em uma das melhores práticas citadas: CobiT – Objetivos de Controle para Tecnologia da Informação e Afins (*Control Objectives for Information and related Technology*).

3.1 COBIT

O CobiT fornece boas práticas através de um modelo de domínios e processos e apresenta atividades em uma estrutura lógica e gerenciável. As boas práticas do CobiT representam o consenso de especialistas. Elas são fortemente focadas mais nos controles e menos na execução. Essas práticas irão ajudar a otimizar os investimentos em TI, assegurar a entrega dos serviços e prover métricas para julgar quando as coisas saem erradas [IT GOVERNANCE INSTITUTE, 2007, p. 8].

Nesse contexto, constata-se que as áreas foco da Governança de TI podem ser representadas de acordo com o exposto na figura 2.

Figura 2 - Áreas de Foco na Governança de TI

Dessa forma, os principais pontos sobre cada área de foco na Governança de TI podem ser descritos da seguinte maneira:

- **Alinhamento estratégico**: foca em garantir a ligação entre os planos de negócios e de TI, definindo, mantendo e validando a proposta de valor de TI, alinhando as operações de TI com as operações da organização.

- **Entrega de valor**: é a execução da proposta de valor de IT através do ciclo de entrega, garantindo que TI entrega os prometidos benefícios previstos na estratégia da organização, concentrando-se em otimizar custos e provendo o valor intrínseco de TI.

- **Gestão de recursos**: refere-se à melhor utilização possível dos investimentos e o apropriado gerenciamento dos recursos críticos de TI: aplicativos, informações, infraestrutura e pessoas. Questões relevantes referem-se à otimização do conhecimento e infraestrutura.

- **Gestão de risco**: requer a preocupação com riscos pelos funcionários mais experientes da corporação, um entendimento claro do apetite de risco da empresa e dos requerimentos de conformidade, transparência sobre os riscos significantes para a organização e inserção do gerenciamento de riscos nas atividades da companhia.

- **Mensuração de desempenho**: acompanha e monitora a implementação da estratégia, término do projeto, uso dos recursos, processo de performance e entrega dos serviços, usando, por exemplo, *balanced scorecards*, que são “cartões” que traduzem as estratégia em ações para atingir os objetivos, medidos através de processos contábeis convencionais.

Dessa forma, para o CobiT conseguir efetivamente apoiar os requisitos de negócio de alto nível de uma empresa, faz-se necessário definir um conjunto genérico de objetivos de negócios e de TI, o que fornece uma base mais refinada para o estabelecimento dos requisitos de negócios e o desenvolvimento de métricas que permitam avaliar se esses objetivos foram atendidos.

Nessa linha de raciocínio, a estratégia da empresa deve ser traduzida pela área de negócios em objetivos relacionados às iniciativas de TI (objetivos de negócios para

TI). Esses objetivos devem levar a uma clara definição dos objetivos próprios da área de TI (os objetivos de TI), o que por sua vez irá definir os recursos e capacidades de TI (a arquitetura de TI para a organização) necessários para executar de maneira bem-sucedida a parte que cabe à TI na estratégia da empresa [IT GOVERNANCE INSTITUTE, 2007, p. 13]. A Figura 3 ilustra esse processo.

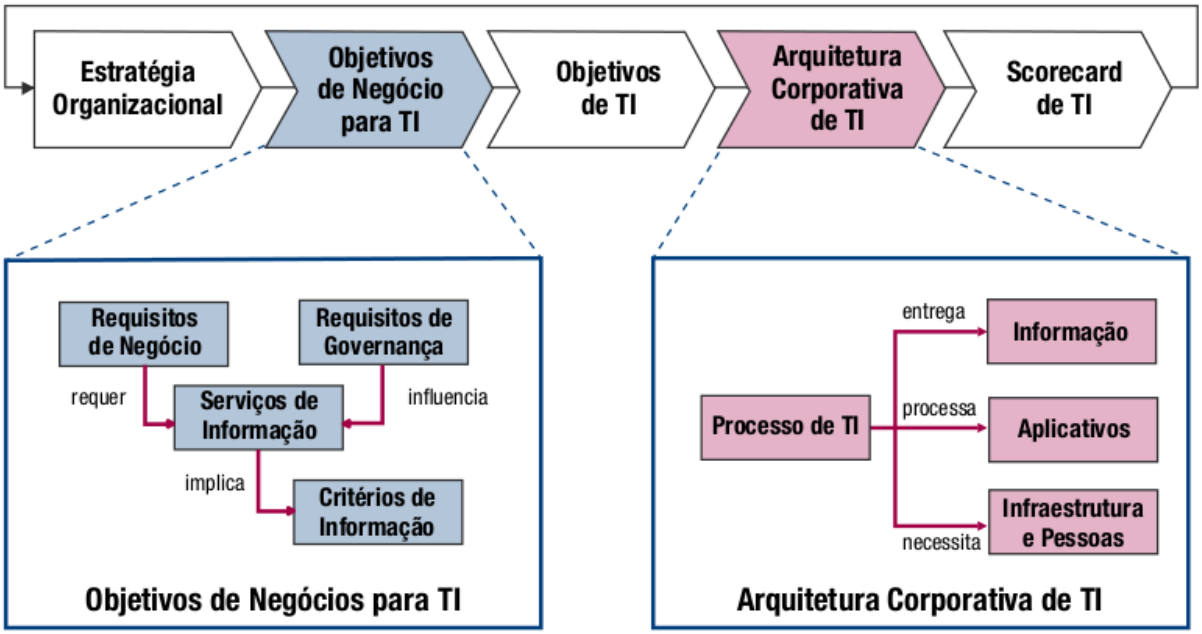


Figura 3 - Definindo os objetivos de TI e a Arquitetura da Empresa para TI

3.1.1 **Orientação a processos**

Com o intuito de organizar e categorizar os diversos processos CobiT que possam ser definidos para um ambiente empresarial, esta ferramenta define as atividades de TI em um modelo de processos genéricos com quatro domínios [IT GOVERNANCE INSTITUTE, 2007, p. 18]. Esses domínios estão representados de forma sintetizada na figura 4 e resumidamente descritos a seguir.

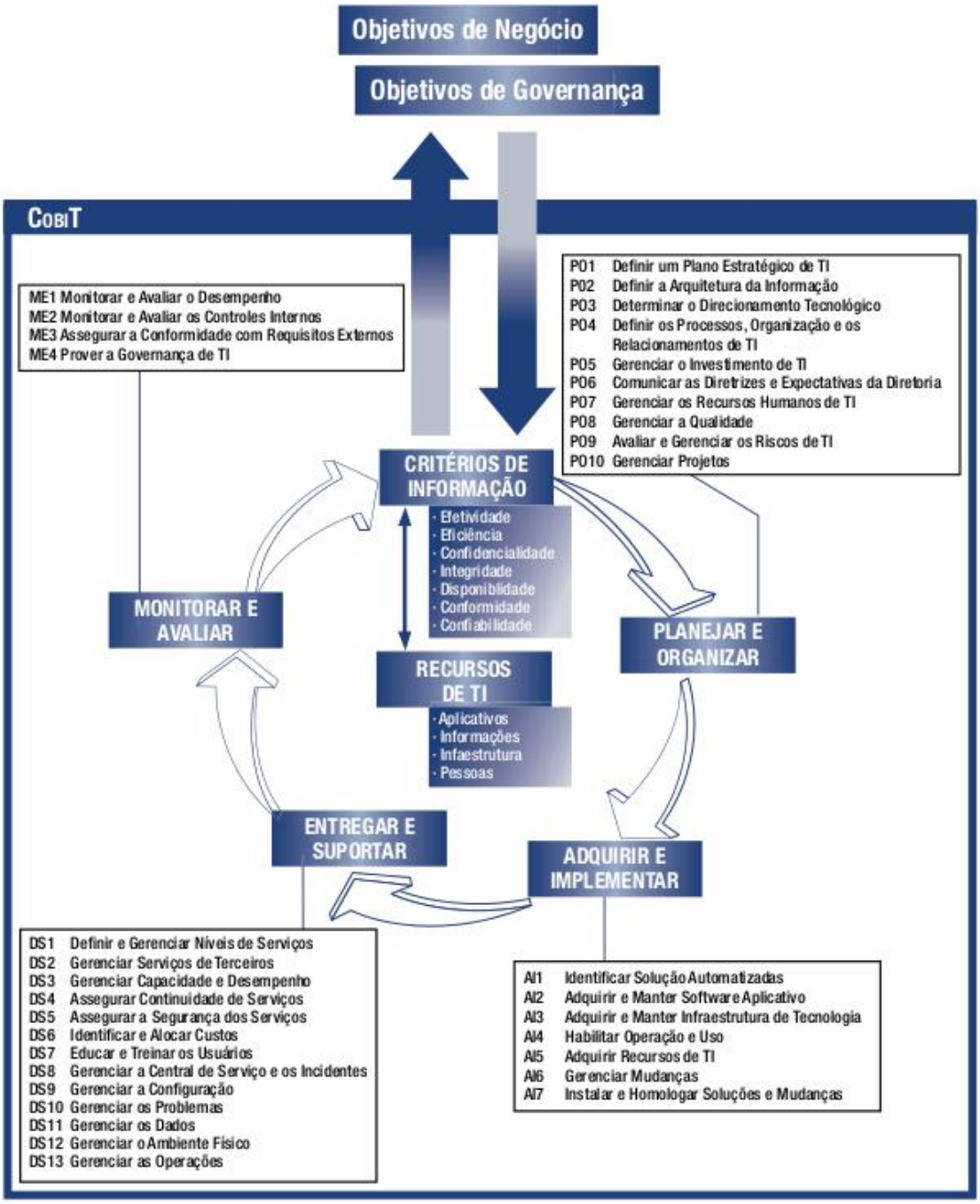


Figura 4 - Visão Geral do Modelo CobiT

- **Planejar e Organizar**: cobre a estratégia e as táticas, preocupando-se com a identificação da maneira em que TI pode melhor contribuir para atingir os objetivos de negócios;

- **Adquirir e Implementar**: Para executar a estratégia de TI, as soluções de TI precisam ser identificadas, desenvolvidas ou adquiridas, implementas e integradas ao processo de negócios;

- **Entregar e Suportar (Deliver and Support)**: trata da entrega dos serviços solicitados, o que inclui entrega de serviço, gerenciamento da segurança e continuidade, serviços de suporte para os usuários e o gerenciamento de dados e recursos operacionais; e

- **Monitorar e Avaliar (Monitor and Evaluate)**: aborda o gerenciamento de performance, o monitoramento do controle interno, a aderência regulatória e a governança.

3.1.2 **Direcionamento Baseado em Medição**

Uma necessidade básica para toda organização é entender a situação dos seus sistemas de TI e decidir que nível de gerenciamento e controle a empresa deveria ter, sempre analisando se o custo é justificado pelo benefício retornado à empresa.

Com esse intuito, o CobiT utiliza um Modelo de Maturidade para o gerenciamento e controle dos processos de TI. Nele, uma definição genérica (Tabela 3) é provida para as escalas de maturidade. Um modelo específico é fornecido derivando dessa escala genérica para cada um dos 34 processos CobiT [IT GOVERNANCE INSTITUTE, 2007, p. 19].

As escalas nesse referido modelo não devem ser tão granulares, visto que seria uma precisão não justificável, pois em geral o propósito é identificar onde estão as questões e como definir prioridades para aprimoramentos.

Tabela 3- Modelo de Maturidade Genérico

Nível	Nome	Descrição
0	Inexistente	A empresa nem mesmo reconheceu que existe uma questão a ser trabalhada
1	Inicial / *Ad Hoc*	A empresa reconheceu que existem questões e que precisam ser trabalhadas, mas não existe processo padronizado, apenas enfoques *Ad Hoc* (aplicados caso-a-caso)
2	Repetível, mas Intuitivo	Procedimentos similares são seguidos por diferentes pessoas fazendo a mesma tarefa. Não existe um treinamento formal ou padronização de processos
3	Definido	Procedimentos padronizados, documentados e comunicados através de treinamento, mas, possivelmente desvios não serão detectados
4	Gerenciado e Mensurável	Monitoramento e medição da aderência aos procedimentos
5	Otimizado	Processos refinados a um nível de boas práticas, baseado em melhoria contínua, tornando a organização rápida em adaptar-se

3.2 CONCLUSÕES

Neste capítulo, definiram-se os principais conceitos sobre Governança de TI e como uma ferramenta como o CobiT, por ser orientado a processos e baseado em medição, colabora para que uma empresa consiga alinhar suas funções de TI com seus objetivos estratégicos de negócio.

Portanto, é de grande importância planejar e executar a TI seguindo padrões de boas práticas como os apresentados no CobiT.

4 **ZABBIX APLICADO NO AMBIENTE COMPUTACIONAL DA FAB SEGUNDO BOAS PRÁTICAS DE GOVERNANÇA DE TI**

Dentre as políticas definidas no âmbito do CCA-RJ, constam Princípios e Diretrizes (PD) relacionadas ao planejamento de TI. O PD12, mais especificamente, trata de “Gerenciar (planejar, organizar, documentar, implementar, medir, acompanhar, avaliar e melhorar) todos os serviços e processos de TI críticos para a organização”, cuja origem está explicitamente definida como CobiT [DTI, 2013].

De fato, a missão de um gerenciador de redes não pode se limitar meramente à parte técnica da atividade, pois este tem uma função mais abrangente que envolve desde a parte técnica propriamente dita até o gerenciamento de pessoal e interfaceamento com os gerentes de negócio.

Sendo assim, este capítulo trata da implantação do Zabbix seguindo alguns dos princípios básicos propostos no CobiT. Para isso, como demonstração da aplicabilidade desta ferramenta neste Trabalho, avaliou-se os processos de seus três primeiros domínios, sendo definido um processo de cada para ser tomado como principal e um ou mais processos tomados como secundários, sobre os quais são feitas algumas observações pertinentes.

4.1 PLANEJAR E ORGANIZAR

Neste domínio, definiu-se o processo PO7 – Gerenciar os Recursos Humanos de TI – como base.

O PO7 consiste formalmente em adquirir, manter e motivar uma força de trabalho competente para criar e entregar serviços de TI para o negócio. Para isso, são seguidos procedimentos definidos de recrutamento, treinamento, avaliação de desempenho e

desligamento [IT GOVERNANCE INSTITUTE, p. 57]

Este processo aborda oito Objetivos de Controle – OC (*Control Objectives – CO*), dentre os quais são tomados alguns para aplicação no contexto deste Trabalho:

- PO7.1 Recrutamento e Retenção de Pessoal;
- PO7.2 Competências Pessoais; e
- PO7.4 Treinamento do Pessoal.

O primeiro processo (PO7.1) assegura que os processos de recrutamento de pessoal estejam alinhados com as políticas e os procedimentos de pessoal da organização.

O segundo processo (PO7.2) verifica regularmente se o pessoal tem as competências necessárias para exercer suas funções com base na formação, no treinamento e/ou na experiência.

No contexto desses dois primeiros processos, o estudo de caso deste Trabalho desenvolver-se-á a partir de quatro militares, que serão referenciados, por medidas de segurança, pelas letras “E”, “I”, “M” e “L”, todos pertencendo ao efetivo da equipe de segurança da informação do CCA-RJ.

Estes militares foram devidamente recrutados de forma alinhada com as políticas de gerenciamento de pessoal do CCA-RJ, bem como foi feita uma constatação prévia da competência destes na área de gerenciamento de redes de computadores. Desta forma, atende-se o PO7.1 e P07.2.

Já o PO7.4, trata de prover ao pessoal de TI treinamento apropriado para manter conhecimento, especializações, habilidades, conscientização sobre controles internos e segurança no nível exigido para atingir os objetivos organizacionais.

Nesse contexto, foram concebidos dois cursos formais:

- Um curso sobre diversas ferramentas (entre elas o Zabbix) para diversos militares da equipe de segurança da informação do CCA-RJ nas dependências deste, dentre os quais o militar “I” participou com sucesso;
- Um curso formal especializado em Zabbix, ministrado na cidade de São Paulo, no qual o militar “L” participou.

Dentre algumas das entradas e saídas em geral para cada processo, o CobiT traz a definição de Tabela RACI, que é formada por atividades e orientações sobre papéis e responsabilidades, sendo indicado quem é responsável (R), responsabilizado (*accountable* - A), consultado (C) e informado (I).

Para o exemplo deste processo PO7 (Gerenciar os Recursos Humanos de TI), tem-se que a tabela RACI é dada pela Tabela 4.

Tabela 4 - Tabela RACI genérica para PO7 - Gerenciar os Recursos Humanos de TI

Atividades	CEO	CFO	Executivo de Negócio	CIO	Proprietário do Processo de Negócio	Responsável por Operações	Responsável por Arquitetura	Responsável por Desenvolvimento	Responsável pela Administração de TI	PMO	Conformidade, auditoria, risco e segurança
Identificar habilidades, descrição de cargos, faixas salariais e comparações de desempenho individual com o mercado *(benchmarks)* para TI;		C		A		C	C	C	R	C	
Executar políticas e procedimentos de RH relevantes para TI (recrutamento, contratação, compensação, treinamento, avaliação, promoção e desligamento)				A		R	R	R	R	R	C

Porém, para o contexto deste Trabalho, será usada uma tabela RACI simplificada para todos os processos abordados em todos os domínios, que pode ser resumida da forma apresentada na Tabela 5.

Tabela 5 - Tabela RACI utilizada para todos os processos CobiT deste Trabalho

Militar	Funções
E	A,C
I	R,C
M	R
L	R

Para o PO7, constatou-se que está contemplado o nível de maturidade “Repetível, porém intuitivo” (nível 2 da Tabela 3), que, traduzido para o contexto específico deste processo, tem-se que há uma abordagem tática na admissão e no gerenciamento do pessoal de TI impulsionada pelas necessidades específicas de projetos. Além disso, realiza-se treinamento informal para o pessoal novo, que a partir de então recebe treinamento somente quando necessário.

A meta nesse contexto é evoluir para, num futuro próximo, seja atingido o nível 3 de maturidade “Processo Definido”, no qual existe um processo definido e documentado para o gerenciamento dos recursos humanos de TI, bem como um programa de reciclagem visando expandir as habilidades técnicas e de gerenciamento de negócio.

4.2 ADQUIRIR E IMPLEMENTAR

No domínio Adquirir e Implementar do CobiT, serão abordados dois processos:

- AI1 – Identificar Soluções Automatizadas, tomado como processo secundário; e
- AI7 - Instalar e Homologar Soluções e Mudanças, como processo principal.

4.2.1 **AI1 - Identificar Soluções Automatizadas**

De acordo com este processo, quando da necessidade de uma nova aplicação ou função, é requisitado uma análise prévia à aquisição ou ao desenvolvimento para assegurar que os requisitos de negócio sejam atendidos através de uma abordagem eficaz e eficiente [IT GOVERNANCE INSTITUTE, p. 75].

Este processo contempla a definição das necessidades, considera fontes alternativas, a revisão de viabilidade econômica e tecnológica.

Quanto aos OCs deste processo, destaca-se para este contexto os seguintes:

- AI1.1 - Definição e Manutenção de Requisitos Técnicos e Funcionais de Negócio;
- AI1.3 - Estudo de Viabilidade e Formulação de Ações Alternativas; e
- AI1.4 - Decisão e Aprovação de Requisitos e Estudo de Viabilidade.

Estes OCs foram atendidos ao longo da Especificação de Requisitos (1.3) no início deste trabalho, do qual podem ser deduzidas as seguintes relações diretas:

- Item 1.3.3 (Tarefa) e AI1.1;
- Item 1.3.5 (Análise de Viabilidade das Alternativas de Solução) e AI1.3; e
- Item 1.3.6 (Definição da Alternativa de Solução Escolhida) e AI1.4.

4.2.2 **AI7 - Instalar e Homologar Soluções e Mudanças**

Este processo, para o âmbito deste Trabalho, constitui um dos mais importantes. No AI, quando se desenvolve um sistema, é necessária uma atenção especial ao colocar este em operação, quando devem ser realizados testes em um ambiente dedicado, com dados de teste relevantes, definição de instruções de implantação e

planejamento de liberação e mudanças no ambiente de produção [IT GOVERNANCE INSTITUTE, p. 99].

Essas tarefas se tornam ainda mais importantes no contexto da implantação de um sistema como o Zabbix, pois este está constantemente passando por mudanças, uma vez que está diretamente atrelado ao surgimento ou desaparecimento de servidores e serviços nos diversos sistemas presentes no CCA-RJ.

Dentre os OCs deste processo, definiu-se os dois seguintes como mais importantes:

- **AI7.3 Plano de Implementação**: Estabelecer um plano de implementação da solução, obtendo para este a aprovação de todas as partes relevantes; e
- **AI7.1 Treinamento**: Treinar a equipe de operações de TI de acordo com o plano de implementação definido.

Primeiramente, após os passos desenvolvidos no processos AI1 – Identificar Soluções automatizadas (4.2.1) e as capacitações realizadas no PO7 - Gerenciar os Recursos Humanos de TI (4.1), foi possível compreender maiores detalhes da arquitetura do Zabbix (2.3) para que fosse possível estudar como seria sua aplicação ao ambiente no CCA-RJ, preparando o terreno para atingir o OC AI7.3 com sucesso.

Quanto aos requerimentos para instalação do servidor Zabbix (2.4), definiu-se que este iria rodar em uma máquina virtual executando em um servidor de virtualização. Esta atende às seguintes características:

- Plataforma Linux (Debian 6.0);
- Processador Intel(R) Xeon(R) 2.40GHz;
- Memória RAM 2GB;

- Banco de dados PostgreSQL 8.4.

Quanto ao caso específico do banco de dados, ao ser definido sua capacidade mínima, analisou-se a fórmula constante no item 2.4.3 utilizando as seguintes estimativas (baseadas em uma série de testes iniciais):

Tabela 6 - Estimativas para Banco de Dados

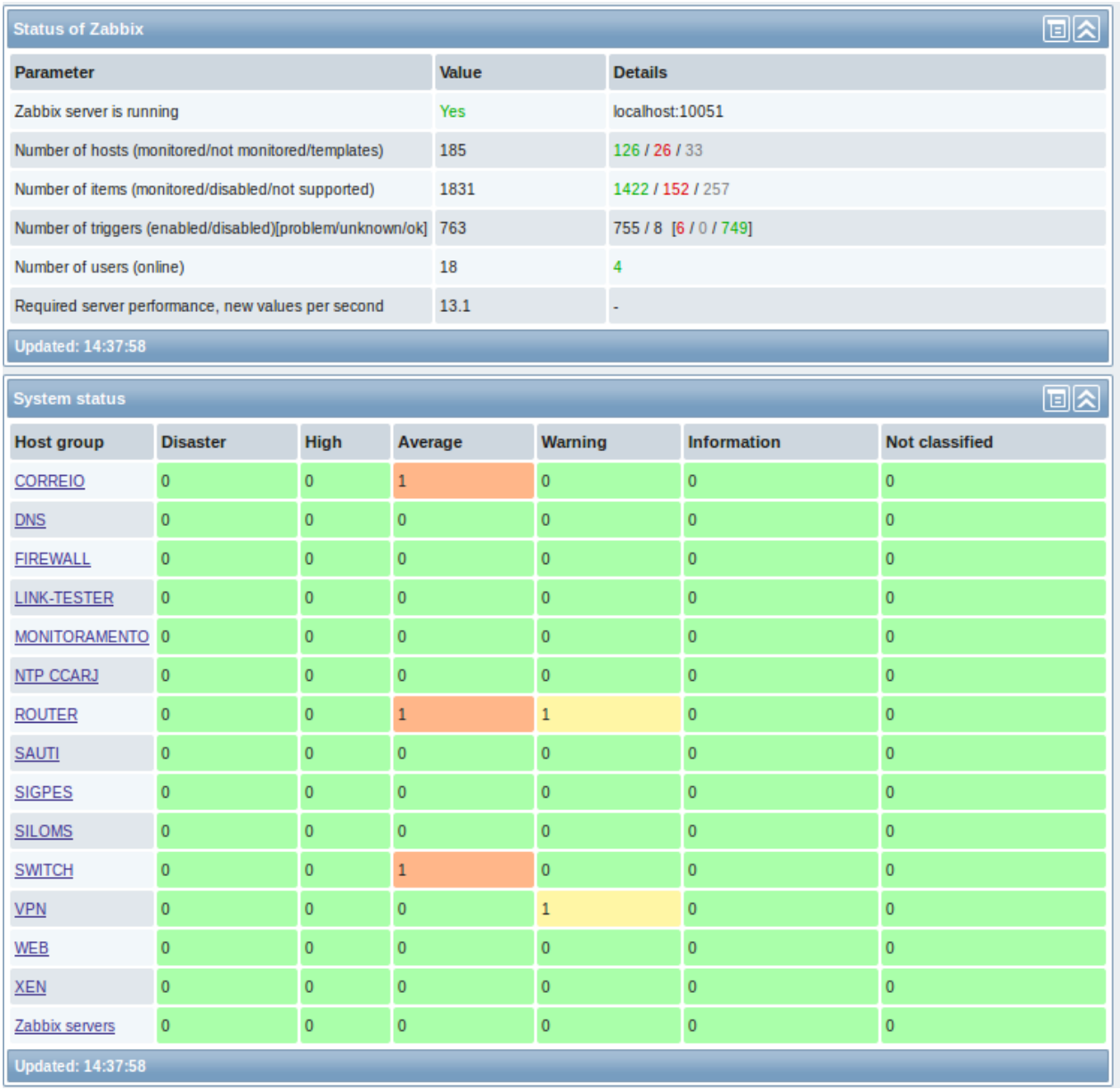
Fator	Fórmula	Valor final
Dias (*trend*)	365	365
Dias (histórico)	90	90
Itens	150*10	1500
Taxa de atualização	300*0.7+30*0.3	219
Eventos	60/(60*3)	0.333
Tam_total	Aproximadamente **5 GB**	

Desta forma, para o PostgreSQL foi reservado, por medidas de segurança, um total de 10GB que, somados aos dados para o Sistema Operacional e dados satélites, totalizou-se um disco de 15GB.

A partir daí, deu-se procedimento a instalação do servidor (de acordo com o item 2.5.1), bem como a dos agentes (item 2.5.2) em algumas máquinas alvo pré-definidas. As máquinas monitoradas (por meio do agente ou não), foram organizadas em grupos lógicos (de acordo com a função destas nos vários sistemas hospedados no CCA-RJ). O quadro geral para esta configuração pode ser observado na figura 5.

Neste quadro geral observa-se algumas informações sobre o *status* do servidor (como número de *hosts*, itens e *triggers*) e o *status* dos diversos sistemas,

apresentando para estes o número de *triggers* que estão com o *status* de problema ativado e sua respectiva gravidade (dividida em uma escala decrescente: *Disaster, High, Average, Warning, Information* e *Not Classified*).



Status of Zabbix

Parameter	Value	Details
Zabbix server is running	Yes	localhost:10051
Number of hosts (monitored/not monitored/templates)	185	126 / 26 / 33
Number of items (monitored/disabled/not supported)	1831	1422 / 152 / 257
Number of triggers (enabled/disabled)[problem/unknown/ok]	763	755 / 8 [6 / 0 / 749]
Number of users (online)	18	4
Required server performance, new values per second	13.1	-

Updated: 14:37:58

System status

Host group	Disaster	High	Average	Warning	Information	Not classified
CORREIO	0	0	1	0	0	0
DNS	0	0	0	0	0	0
FIREWALL	0	0	0	0	0	0
LINK-TESTER	0	0	0	0	0	0
MONITORAMENTO	0	0	0	0	0	0
NTP CCARJ	0	0	0	0	0	0
ROUTER	0	0	1	1	0	0
SAUTI	0	0	0	0	0	0
SIGPES	0	0	0	0	0	0
SILOMS	0	0	0	0	0	0
SWITCH	0	0	1	0	0	0
VPN	0	0	0	1	0	0
WEB	0	0	0	0	0	0
XEN	0	0	0	0	0	0
Zabbix servers	0	0	0	0	0	0

Updated: 14:37:58

Figura 5 - *Dashboard* geral para o Zabbix no CCA-RJ

As *triggers* desenvolvidas para os itens no Zabbix analisam os mais diversos tipos de problemas de disponibilidade e performance. Dentre os mais comuns, podem

ser citados questões de saturação de *links*, conforme apresentado na figura 6.

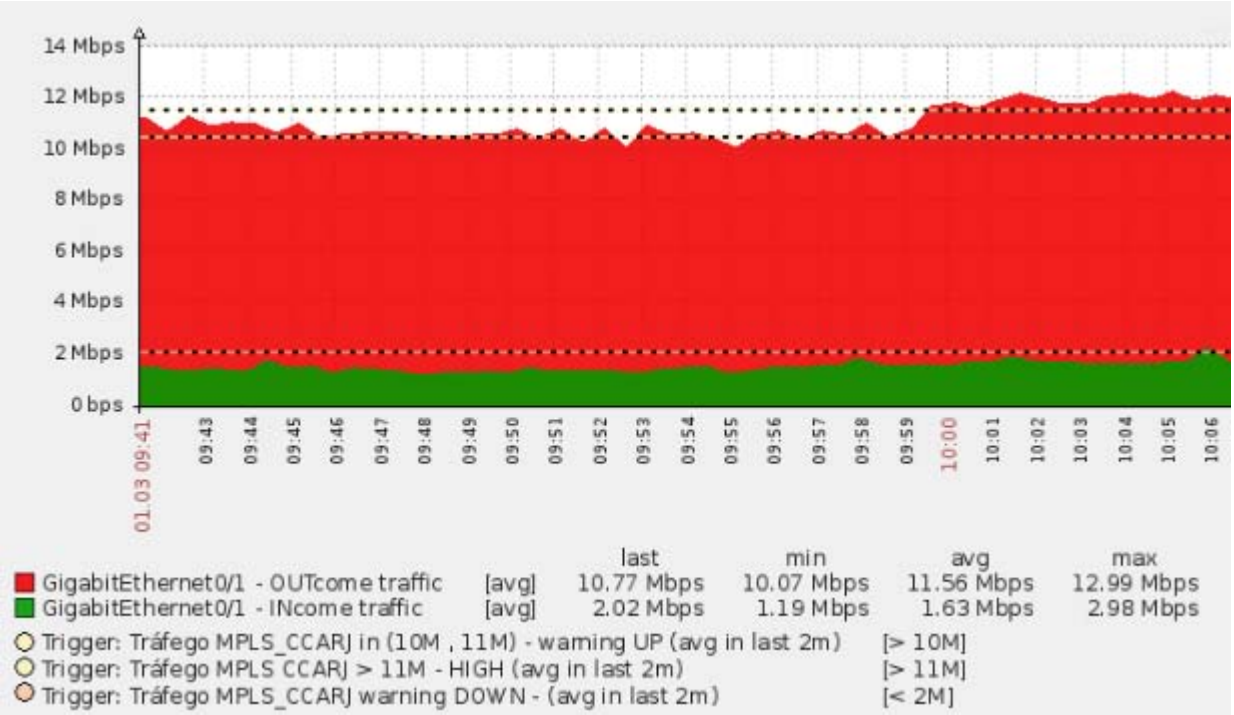


Figura 6 - Tráfego em um dos *links* de INTRAER do CCA-RJ

Dessa forma, atingiu-se com sucesso o OC AI7.3 (Plano de Implementação) de forma que os procedimentos estabelecidos neste foram devidamente documentados e repassados à equipe de operações de TI, formada pelos militares “M” e “L” e chefiadas pelo militar “I”, de forma que se atingiu também com sucesso o OC AI7.1 (Treinamento).

Vale ressaltar que, por medida de segurança, são omitidos neste Trabalho alguns detalhes que possam vir a comprometer a estrutura dos sistemas internos da FAB.

4.3 ENTREGAR E SUPORTAR

No contexto deste domínio, são aplicados dois processos:

- **DS10 – Gerenciar Problemas**: definido como processo principal neste

domínio, trata da identificação e classificação dos problemas, análise de causas-raiz e respectiva resolução. O processo de gerenciamento de problemas também contempla a identificação de recomendações para melhoria, manutenção dos registros de problemas e revisão da situação das ações corretivas.

- **DS11 – Gerenciar os Dados**: definido como processo secundário, contempla o estabelecimento de procedimentos efetivos para controlar a biblioteca de mídia, cópia de segurança (*backup*), recuperação de dados e a dispensa de mídias de forma adequada.

4.3.1 **DS10 – Gerenciar Problemas**

No contexto do processo de gerenciamento de problemas, deu-se enfoque em dois OCs principais:

- **DS10.1 Identificar e Classificar os Problemas**: desenvolve processos para reportar e classificar os problemas identificados; e
- **DS10.2 Rastreamento e Resolução de Problemas**: fornece recursos que permitam o rastreamento, a análise e a identificação da causa-raiz dos problemas reportados.

Os passos envolvidos na classificação de problemas são similares aos passos da classificação de incidentes; eles servem para determinar a categoria, o impacto, a urgência e a prioridade.

Porém, uma das vantagens do Zabbix é que, uma vez que o *trigger* para um item tenha sido bem definido (por exemplo: “o espaço livre no disco de armazenamento do banco de dados principal está menor do que 1% gera um alerta nível *high*”), quando da

ocorrência do problema descrito, este será automaticamente identificado e classificado, conforme as categorias já apresentadas na figura 5. Sendo assim, o OC DS10.1 é atendido no contexto do Zabbix de forma automatizada.

Quanto ao OC DS10.2, foi desenvolvido e documentado um procedimento para ser utilizado no rastreamento e resolução de problemas que tenham sido classificados. Este procedimento encontra-se apresentado de forma resumida na figura 7, em que detalhes adicionais foram emitidos por fugirem ao escopo deste Trabalho.

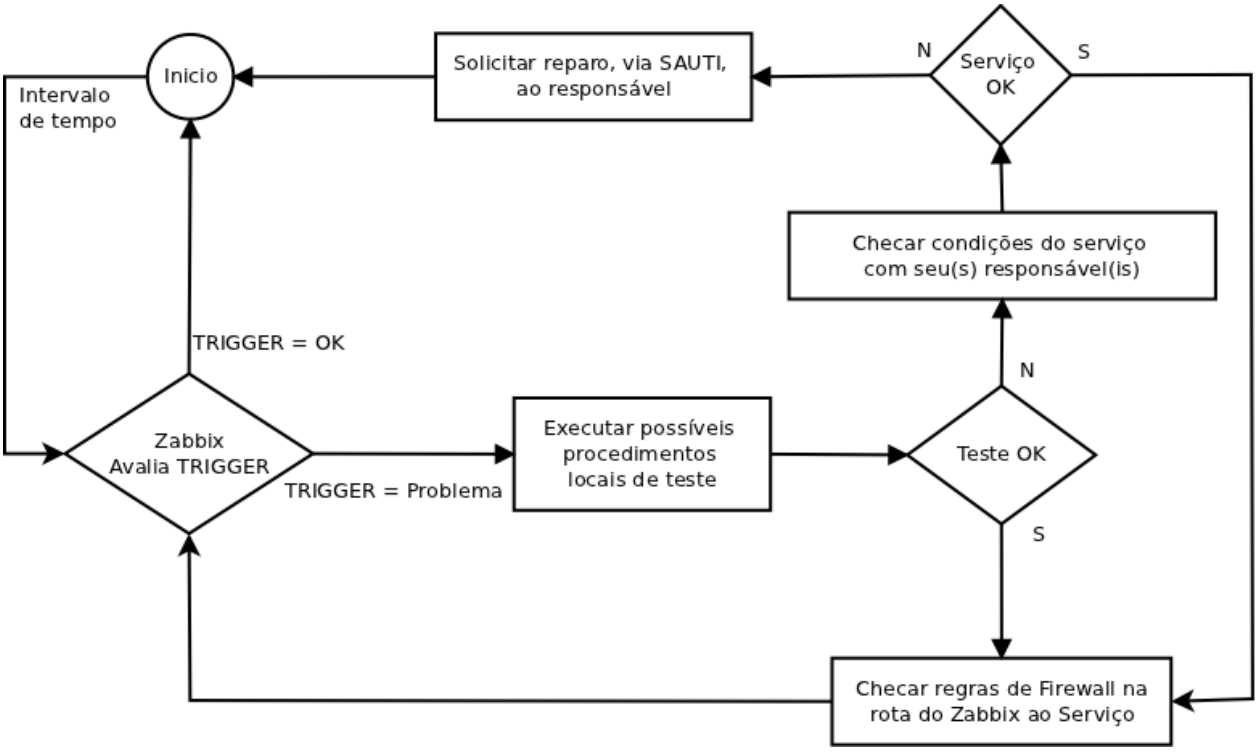


Figura 7 - Fluxograma para Rastreamento e Resolução de Problemas

Quanto ao nível de maturidade neste processo, tem-se que se atingiu o nível 3 (Tabela 3), pois a resolução de problemas e os processos de encaminhamento foram padronizados. Além disso, os registros, rastreamento e resoluções de problemas são fragmentados dentro da equipe, utilizando as ferramentas disponíveis sem

centralização.

4.3.2 **DS11 – Gerenciar os Dados**

O processo de gerenciamento de dados, dentre suas metas, contempla o estabelecimento de procedimentos efetivos para controlar cópia de segurança (*backup*) e recuperação de dados.

Este propósito está diretamente ligado a um dos OC do processo DS11: “DS11.5 Backup e Restauração”. No intuito de atender este OC, foi desenvolvida a seguinte política de *backup* do servidor:

- Intervalo de 15 dias entre cada *backup*;
- Rotação com 4 *backups*, ou seja, sempre haverá apenas 4 cópias armazenadas. No caso de esse número chegar a 5, o *backup* mais antigo é removido;
- Cópia realizada em um servidor remoto em outra rede;
- Procedimento como um todo realizado de forma automatizada, por meio do *script* disponível no Anexo 3.

A figura 8 representa o resultado do *backup* remoto do servidor Zabbix.

```
root@zabbix:/backuping# ls -R /mnt/remote_bkp_path/zabbix/
/mnt/remote_bkp_path/zabbix/:
1  2  3  4

/mnt/remote_bkp_path/zabbix/1:
FULL_BACKUP  snapshot  summary_11-03-13.log  tared_files.log  zabbix.tar.bz2

/mnt/remote_bkp_path/zabbix/2:
FULL_BACKUP  snapshot  summary_01-03-13.log  tared_files.log  zabbix.tar.bz2

/mnt/remote_bkp_path/zabbix/3:
FULL_BACKUP  snapshot  summary_21-02-13.log  tared_files.log  zabbix.tar.bz2

/mnt/remote_bkp_path/zabbix/4:
FULL_BACKUP  snapshot  summary_11-02-13.log  tared_files.log  zabbix.tar.bz2
```

Figura 8 - Resultado do *backup* remoto do Zabbix

5 CONCLUSÕES

De acordo com o exposto neste trabalho, constatou-se que é nítida a necessidade de se implantar um sistema eficiente de monitoramento e gerenciamento de serviços de rede em qualquer grande empresa que dependa do uso de redes interconectadas de computadores.

Dentre as opções possíveis para o gerenciamento, definiu-se o Zabbix devido à sua melhor adequabilidade aos requisitos necessários no contexto particular da FAB. Porém, constatou-se também que, para uma efetiva solução, este sistema deveria ser implantado seguindo princípios das boas práticas na área de Governança de TI, para garantir o alinhamento da equipe de TI (nível operacional) com área de negócios (alto nível).

Portanto, foram selecionados alguns processos do CobiT para orientar um estudo de caso de implantação do Zabbix no ambiente de operação real da FAB, mais especificamente no CCA-RJ. Tal estudo foi concluído com sucesso e serviu fortemente de base para a implantação real no ambiente operacional de fato do CCA-RJ.

REFERÊNCIAS

OLUPS, Richard. **Zabbix 1.8 Network Monitoring**. Birmingham, UK : Packt Publishing Ltd. 2010. 428p.

DTI. **ICA 37-449:** Plano de Unidades Didáticas do Curso de Manutenção de Rede Local Avançado (CMRLA). Portaria DTI Nº 15/SDAD, de 23 de março de 2011. 13p.

___. **PCA 7-7:** Plano Diretor de Tecnologia da Informação do Centro de Computação da Aeronáutica do Rio de Janeiro (CCA-RJ) – período de 2013 a 2014. Portaria DTI Nº 01/AGTI, de 03 de janeiro de 2013. 34p.

___. **Rica 21-236:** Regimento Interno da Diretoria de Tecnologia da Informação da Aeronáutica. Portaria COMGAP Nº 8/3EM, de 28 de março de 2011. 47p.

Hackers invadem e tiram site da FAB do ar. **ESTADAO**, São Paulo, 25 nov. 2008. Disponível em: < http://www.estadao.com.br/noticias/tecnologia,hackers-invadem-e-tiram-site-da-fab-do-ar,283008,0.htm>. Acesso em: 25 jul. 2012

IT GOVERNANCE INSTITUTE. **Cobit 4.1**: Modelo, Objetivos de Controle, Diretrizes de Gerenciamento e Modelos de Maturidade. 2007. 212p.

NAGIOS. **Nagios Overview.** 2013. Disponível em: < http://www.nagios.org/about/overview/ >. Acesso em: 8 fev. 2013

SERPRO. **Governança de TI**. 2013. Disponível em: < https://www.serpro.gov.br/conteudo-tecnologia/desenvolvimento/governanca-de-ti>. Acesso em: 10 fev. 2013.

ZABBIX SIA. **Zabbix Manual**. 2013. Disponível em: <https://www.zabbix.com/documentation/2.0/manual>. Acesso em: 14 fev. 2013.

ANEXOS

1. SEQUENCIA DE PASSOS PARA INSTALAÇÃO DO SERVIDOR ZABBIX

1 – Na página de downloads oficial Zabbix download page(http://www.zabbix.com/download.php),
baixar os arquivos-fonte para /usr/local/src e extraí-los:

```
$ cd /usr/local/src
$ tar -zxvf zabbix-2.0.0.tar.gz
```

2 – Configurando permissionamento:

```
groupadd zabbix
useradd -g zabbix zabbix
```

3 – Instalando o Banco de Dados:

```
apt-get install postgresql-8.4

su postgres
createdb zabbix
createuser -sEP zabbix

cd database/postgresql
su zabbix
psql -U zabbix zabbix < schema.sql
# stop here if you are creating database for Zabbix proxy
psql -U zabbix zabbix < images.sql
psql -U zabbix zabbix < data.sql
```

4 – Preparando o ambiente para a compilação do código-fonte

```
apt-get install apache2 php5  gcc make
apt-get install postgresql-server-dev-8.4 libcurl4-openssl-dev snmp libsnmp-dev
./configure  --enable-server  --enable-agent  --with-postgresql  --with-net-snmp  --with-libcurl
```

5 – Compilando e instalando

```
make
make install
```

6 – Arquivos de configuração

substituir os arquivos conf no diretório de configuração (zabbix_agent.conf zabbix_agentd zabbix_agentd.conf zabbix_server.conf zabbix_serverd) nos respectivos locais

7 – Habilitando para execução automática sistemática

```
update-rc.d zabbix_serverd defaults
update-rc.d zabbix_agentd defaults

root@zabbix:/usr/local/etc# touch /var/log/zabbix_server.log
root@zabbix:/usr/local/etc# vim zabbix_agentd.conf
root@zabbix:/usr/local/etc# touch /var/log/zabbix_agentd.log
root@zabbix:/usr/local/etc# chown zabbix:zabbix /var/log/zabbix_server.log
root@zabbix:/usr/local/etc# chown zabbix:zabbix /var/log/zabbix_agentd.log


8 – Arquivos PHP (em frontends/php)

mkdir /var/www/zabbix
cd frontends/php
cp -a . /var/www/zabbix
chown -R zabbix:www-data /var/www/zabbix/

9 – Instalando extensões necessárias ao php5:

apt-get install php5-pgsql php5-suhosin


10 – Instalando o Frontend:
A partir daqui, basta apenas acessar a URL http://<server_ip_or_name>/zabbix  e seguir
o passo a passo sugerido na interface web e concluir a instalação do servidor.
```

2. *SCRIPT* DE INSTALAÇÃO DO AGENTE ZABBIX EM MÁQUINAS DEBIAN-LIKE

```
#!/bin/bash

# ---------------- DEPENDENCIAS ----------------------------- #
depend_vet=("gcc" "make")

for depend in ${depend_vet[@]}; do
      command -v $depend > /dev/null 2>&1 || { echo "Programa \"$depend\" nao existe.
Se deseja continuar, instale-o (apt-get install $depend) antes de continuar. Abortando
instalacao."; exit 1; }
done

echo "Dependencias ok!"
echo

# ---------------------- CRIANDO O USUARIO ZABBIX --------------------------
echo "Adicionando usuario de sistema 'zabbix':"
echo "adduser --system --no-create-home zabbix"
adduser --system --no-create-home zabbix
echo


# ---------------------- COMPILANDO O CODIGO-FONTE  --------------------------
agent_dir=agent_package

cd $agent_dir

./configure --enable-agent --enable-static
make install

cd ..

# ---------------------- SOBRESCREVENDO OS ARQUIVOS .CONF  E .LOG----------------------
----
echo "Copiando arquivos de configuracao para suas respectivas pastas:"
echo "cp zabbix_agent*.conf /usr/local/etc/"
cp zabbix_agent*.conf /usr/local/etc/
echo

echo "Criando arquivo de log e configurando permissionamento:"
echo "touch /var/log/zabbix_agentd.log"
touch /var/log/zabbix_agentd.log
echo "chown zabbix /var/log/zabbix_agentd.log"
chown zabbix /var/log/zabbix_agentd.log
echo


#----------------------- CONFIGURANDO O INICIALIZADOR DA DAEMON AGENTE ZABBIX --------
-----
chmod +x zabbix_agentd
echo "Configurando o inicializador do agente ZABBIX:"
echo "cp zabbix_agentd /etc/init.d"
cp zabbix_agentd /etc/init.d
echo "update-rc.d -f zabbix_agentd remove"
update-rc.d -f zabbix_agentd remove
echo "update-rc.d zabbix_agentd defaults"
update-rc.d zabbix_agentd defaults
echo
```

```
echo
echo "AGENTE ZABBIX INSTALADO COM SUCESSO."
echo

echo "INICIALIZANDO O AGENTE [/etc/init.d/zabbix_agentd start]:"
/etc/init.d/zabbix_agentd start
echo

#echo "CHECANDO O STATUS DO AGENTE:"
#sleep 1
#/etc/init.d/zabbix_agentd status
#echo

#------------------------ ATUALIZACAO DO SERVIDOR ZABBIX ----------------------
#echo "Comunicando a SMS (Secao de Monitoramento de Seguranca) da instalacao do agente
ZABBIX"
echo
echo "ATENCAO!!!!!!!!!!  Enviar um email para _sms@ccarj.intraer com:

- nome da maquina onde foi instalado o agente [ $(hostname) ]
- ip (se houver NAT, fornecer o ip de NAT)
- Necessidade de checar OU NAO parametros específicos importantes

- Quaisquer duvidas ou sugestoes"
echo

exit 0
```

3. *SCRIPT* DE BACKUP AUTOMATIZADO DO SERVIDOR ZABBIX

```
#=====================================================================
#      INSTRUCOES GERAIS
#=====================================================================

#      Definicoes utilizadas:
#      - fpath = file_path -> descreve o caminho inteiro ateh o arquivo, com o nome
deste inclusive
#      - fname = file_name -> apenas o nome do arquivo, sem seu dirname (nome do seu
diretorio)


#=====================================================================
#       CONSTANTES GLOBAIS
#=====================================================================
readonly current_script_path=$(dirname ${BASH_SOURCE[0]})
initial_pwd=$(pwd)

readonly exclude_fpath=$current_script_path"/exclude_file"              #  arquivo  que
contem os diretorios a serem retirados do backup
readonly include_fpath=$current_script_path"/include_file"              #  ARQUIVO  QUE
CONTÉM OS ARQUIVOS A SEREM BACKED UP
readonly snapshot_fname="snapshot"                    # arquivo que gurada o snapshot do
backup full

readonly summary_log_fpath=$current_script_path"/summary_$(date +%d-%m-%y).log"
       # arquivo de log apenas deste processo de log
readonly tared_files_log_fpath=$current_script_path"/tared_files.log"   #  arquivo  com
o registro dos arquivos que foram "tared"
readonly global_log_fpath="/tmp/bkp.log"              #  arquivo  de  log  global,  onde
estao acumulado os ultimos logs. NÃO é criado automaticamente.

readonly server=$(hostname)                           #nome  do  servidor,  para  que  se
crie a pasta para backup com seu nome
readonly remote_bkp_path="/mnt/remote_bkp_path"            #caminho   de   montagem   do
samba
readonly bkp_tar_name=$server".tar.bz2"               #  nome  padrao  do  arquivo  target
(alvo) comprimido pelo tar

// VARIÁVEIS OMITIDAS DESTE TRABALHO POR QUESTAO DE SEGURANÇA //

readonly nfull=4                                 # numero de backups full
readonly ndiff=0                           # numero de backups diferencial
readonly total_sets=$(( $nfull*($ndiff+1) ))          # ultimo set de backups

readonly full_tag="FULL_BACKUP"                       # TAG QUE INDICA FULL
readonly diff_tag="DIFERENTIAL_BACKUP"                # TAG QUE INDICA DIFERENCIAL


#=====================================================================
#       VARIAVEIS GLOBAIS
#=====================================================================
bkp_tag=""                                       # TAG QUE INDICARAH QUAL O TIPO DE BACKUP
A SER FEITO


#=====================================================================
#       FUNCOES
#=====================================================================

#=====================================================================
```

```
#       ROTINAS CUSTOMIZAVEIS PARA CADA TIPO DE BACKUP
#=====================================================================

## ----- DENTRE SEUS OBJETIVOS, PODEM ESTAR DEFINIR ARQUIVOS DENTRO DE UM DIRETORIO E -
-
## ----- INSERIR O PATH DESSE DIRETORIO NO ARQUIVO DE INCLUDE DO TAR    ---------------
-
pre_backup () {
       tmp_backup_dir=$(./dump_db_and_www.sh)
       echo $tmp_backup_dir > $include_fpath

       return 0
}

## DESFAZER AS CONDICOES TEMPORARIAS CRIADAS PARA O BACKUP
post_backup () {
       rm -Rf $tmp_backup_dir
}
# ------------------------------------------------------------------------------------

#=================================================
#       FUNCOES SMB
#=================================================
smb_is_mounted () {
        local is_mounted=$( df -h | grep -c $remote_bkp_path )
       echo $is_mounted
}

mount_smb () {
        mount      -t      cifs      //$smb_ip/$smb_share      $remote_bkp_path      -o
username=$smb_account,password=$smb_passwd,workgroup=$smb_workgroup                 2>>
$summary_log_fpath
}

umount_smb () {
       # TENTA DESMONTAR O COMPARTILHAMENTO SMB NO MÁXIMO 10x
       for loop in {1..10}; do
              umount $remote_bkp_path
                 sleep 0.5

              if (( ! `smb_is_mounted` )); then
                     # A partir desta linha, essa função irá retornar para a função
principal, sem avançar
                     echo "Compartilhamento SMB desmontado com sucesso"
                     return 0
              fi
         done

         # SE A FUNÇÃO CHEGOU ATÉ AQUI, É SINAL QUE O SMB CONTINUOU MONTADO
         echo "$(date +%d-%m-%y_%H:%M:%S) - ERRO - Script de LOG " "Nao foi possivel
desmontar o compartilhamento samba do script para backup

Verifique:
- Se o sistema de arquivos esta ocupado com algum processo de copia ainda nao concluido
- Se o firewall esta bloqueando a origem ate o destino pelas portas 443 e 139
- Se houve alguma falha na rede que resultou neste acontecimento

Favor, entre no servidor que este scrip esta sendo executado e desmonte o
compartilhamento manualmente com o comando:

# umount $remote_bkp_path" > "error_umountig_sharing$(date +%d-%m-%y_%H:%M:%S).log"
}
```

```
try_mount_smb () {
        if (( ! `smb_is_mounted` )); then
                mount_smb 2>> $summary_log_fpath
        fi

        if (( ! `smb_is_mounted` )); then
                log "Nao foi possivel montar o compartilhamento samba do script para
backup

Verifique:
- Se a conta e senha do compartilhamento esta correta
- Se o firewall esta bloqueando a origem ate o destino pelas portas 443 e 139
- Se o script esta com a configuracao correta para montagem do compartilhamento
- Se a configuracao de montagem contida no script (parte mount -cifs) esta correta"
                log "Saindo do script"
                #sai do script
                exit 1
        fi
}

#=====================================================
#       FUNCOES LOG
#=====================================================
log () {
       msg2log=$1

       if [ $# = 2 ]; then
              file2log=$2
       else
              file2log=$summary_log_fpath
       fi

       if [ $# = 1 ]; then
               echo "$(date +%d-%m-%y_%H:%M:%S) --> $msg2log" >> $file2log

       elif [ $# = 2 ]; then
               echo  "$(date  +%d-%m-%y_%H:%M:%S)  -->  ARQUIVOES  tared  ---------"  >>
$file2log
               echo "$msg2log" >> $file2log
               echo "-----------------------------------------------" >> $file2log
       fi
}

conclude_logging () {
       show_credits >> $summary_log_fpath

       if [ -f $global_log_fpath ]; then
              cat $summary_log_fpath >> $global_log_fpath
       fi

       # anexando os log deste set ao set global
       mv -f $summary_log_fpath $remote_bkp_path/$server/1/
       mv -f $tared_files_log_fpath $remote_bkp_path/$server/1/
}

#=====================================================
#       FUNCOES GERENCIA DE SETS
#=====================================================
ensure_server_folder() {
       # garante que a pasta com o nome do servidor existirah
       if [ ! -d $remote_bkp_path/$server ]; then
               log "Nao existia pasta [$server] no diretorio de backup. Criando esta
pasta."
```

```
                mkdir -p $remote_bkp_path/$server 2>> $summary_log_fpath
        fi
}

remove_last_set () {
        #se existir o ultimo set de backup, remove-o
        if [ -d $remote_bkp_path/$server/$total_sets ]; then
                # CONTROLA PARA QUE UM SET FULL NÃO SEJA DESCARTADO ENQUANTO HAJA
DIFERENCIAL(IS) PRESENTE NA ROTAÇÃO
                if [ -f $remote_bkp_path/$server/$total_sets/$full_tag -a -f
$remote_bkp_path/$server/$(( $total_sets - 1))/$diff_tag ]; then
                        log "Evitando remoção de FULL set, pois existiriam DIFF sets
orfãos. Removendo o DIFF set anterior ao ultimo FULL set."
                        rm -rf $remote_bkp_path/$server/$(( $total_sets - 1 )) 2>>
$summary_log_fpath
                else
                        log "Removendo set de backup [$total_sets], por ser muito antigo
antigo"
                        rm -rf $remote_bkp_path/$server/$total_sets 2>>
$summary_log_fpath
                fi
        fi
}


#rotaciona as pastas de sets de backup
rotate_sets () {
        local current_set=$(( $total_sets - 1 ))

        log "Inicio de Rotacao de SETs"
        for (( ; current_set >= 1; current_set-- ))
        do
                if [ -d $remote_bkp_path/$server/$current_set ]; then
                        # CONTROLA PARA QUE UM SET FULL NÃO SEJA DESCARTADO ENQUANTO HAJA
DIFERENCIAL(IS) PRESENTE NA ROTAÇÃO
                        #if [ $current_set -eq $(( $total_sets - 1 )) -a -f
$remote_bkp_path/$server/$(( $current_set + 1 ))/$full_tag -a -f
$remote_bkp_path/$server/$current_set/$diff_tag ]; then
                        #       log "Evitando overwriting de FULL set, pois existiriam DIFF
sets orfãos."
                        #       continue
                        #fi

                        log "Renomeando set [$current_set] para [$(( $current_set + 1
))]"
                        mv -f $remote_bkp_path/$server/$current_set
$remote_bkp_path/$server/$(( $current_set + 1 )) 2>> $summary_log_fpath
                fi
        done
        log "Fim de Rotacao de SETs"
}

ensure_first_set () {
        # garante que a pasta do primeiro set irah existir
        if [ ! -d $remote_bkp_path/$server/1 ]; then
                log "Nao existia o primeiro set, criando tal"
                mkdir -p $remote_bkp_path/$server/1 2>> $summary_log_fpath
        fi
}

# localiza o set mais recente que eh um backup full.
locate_newer_full_set () {
        local current_set=2
```

```
	for (( ; current_set <= $(( $ndiff + 1 )); current_set++  ))
	do
		if [ -f $remote_bkp_path/$server/$current_set/$full_tag ]; then
			break
		fi
	done

	echo $current_set
}

#=====================================================
#	FUNCOES GERENCIA DE BACKUP (etapa tar)
#=====================================================
# FUNCAO QUE DEFINE A TAG A SER IMPRESSA NO SET ATUAL E IMPORTA, NO CASO CASO DE BACKUP
DIFERENCIAL, O SNAPSHOT REMOTO.
define_snapshot_and_tag () {
	local newer_full_set=$(locate_newer_full_set)

	log "FULL SET mais recente encontrado na posicao [$newer_full_set]"

	if (( $newer_full_set < $(( 2 + $ndiff )) )); then
		log "Decidido por $diff_tag relativo ao SET [$newer_full_set]. Importando
o snapshot deste."
		cp		$remote_bkp_path/$server/$newer_full_set/$snapshot_fname
$snapshot_fname  2>> $summary_log_fpath
		bkp_tag=$diff_tag
	else
		log "Decidido por $full_tag"
		bkp_tag=$full_tag
	fi
}

do_backup () {

	pre_backup

	if [ -f $exclude_fpath  ]; then
		exclude_opt="--exclude-from=$exclude_fpath"
	fi

	log "Realizando o comando \"tar -cjvf $remote_bkp_path/$server/1/$bkp_tar_name
$exclude_opt -g $snapshot_fname --files-from=$include_fpath\""
	log  "$((tar  -cjvf  $remote_bkp_path/$server/1/$bkp_tar_name  $exclude_opt  -g
$snapshot_fname --files-from=$include_fpath) 2>&1)" $tared_files_log_fpath

	log "Exportando snapshot para o destino"
	mv -f $snapshot_fname $remote_bkp_path/$server/1 2>> $summary_log_fpath

	log "Imprimindo a tag $bkp_tag, que representa o tipo de backup"
	touch $remote_bkp_path/$server/1/$bkp_tag 2>> $summary_log_fpath

	post_backup
}


#log "#============================================================="
#log "#		INICIO DA FUNCAO PRINCIPAL	"
#log "#============================================================="
##iniciando rotina de backup

if (( $use_smb_share )); then
	try_mount_smb
```

```
fi

cd $current_script_path

ensure_server_folder

remove_last_set
rotate_sets
ensure_first_set

## O TIPO DE BACKUP A SER FEITO EH DEFINIDO PELO ARQUIVO DE SNAPSHOT
define_snapshot_and_tag

do_backup

#
##====================================================================
##       BACKUP - ROTINA
##====================================================================
#
log "Rotina de backup concluida"
log "============================================================="
#
##====================================================================
##       TERMINO DO SCRIPT - DESMONTAGEM E LIMPEZA
##====================================================================
conclude_logging

if (( $use_smb_share )); then
       echo "Desmontando samba"
       umount_smb
fi

echo
echo Script finalizado com sucesso.
echo

cd $initial_pwd

exit 0
```